Omega3P Top-Level Commands

- **ModelInfo**
- **FiniteElement**
- **PRegion**
- **EigenSolver (Linear & Quadratic)**
- **Nonlinear EigenSolver**
- **Port**
- **Absorbing Boundary Condition (ABC)**
- **PostProcess**

Note: Refer to **acdtool** command syntax for postprocessing capabilities.

ModelInfo

```
ModelInfo :
{
 File: ./pillbox+recWG4.ncdf

 BoundaryCondition : {
  Magnetic:  1, 2
  Electric:  3, 4
  Exterior:  6
  Waveguide: 7
 }

  SurfaceMaterial : {
  ReferenceNumber: 6
  Sigma: 5.8e7
 }

 Material : {
   Attribute: 1
   Epsilon:   1.0
   Mu:        1.0
 }

Material : {
   Attribute: 2
   PML: {
     Type: Face
     BoundaryID: 6
     Thickness: 0.02
     Start: -0.01143 -0.00508 0.08
     End:    0.01143  0.00508 0.10
     Direction: 0. 0. +1.
     Reflection Coefficient:  1.e-5
     Polynomial Order: 2
     Maximum Sigma: 2.e0 2.e0 2.e0
   }
  }

Material : {
  Attribute: 2
  PML: {
    Type: Face
    BoundaryID: 6
    Thickness: 0.004
    SubType: Sphere
```

```
    Radius: 0.05
    Direction: 0. 0. +1.
    Reflection Coefficient:  1.e-5
    Polynomial Order: 2
  }
 }

}
```

ModelInfo specifies the model used for omega3p simulation.

File: The name of the mesh file in netcdf format. It can be located in a directory specified by the path. If the path is not specified, the default is the directory where the job is submitted.

BoundaryCondition specifies the boundary conditions on all the surfaces of the mesh. Every single surface of the mesh has a reference number which is set in Cubit and each reference number is associated with a boundary condition. The types of boundary condition used in omega3p are

- **Electric**: The tangential component of the electric field is zero. It can be used to define a symmetry plane in the model.
- **Magnetic**: The tangential component of the magnetic field is zero. It can be used to define a symmetry plane in the model.
- **Exterior**: The tangential component of the electric field is zero. Computationally it is equivalent to **Electric** boundary condition, but used normally for perfectly conducting cavity surface. For normal conducting surface, it can be used for calculating power loss on the surface using perturbation theory.
- **Impedance**: The electric and magnetic fields are related by the conductivity of the boundary surface for imperfect conductor; useful for self-consistent field calculation for finite loss of a conductor when the skin depth is small compared with the finite element sizes in the mesh. The conductivity is specified in **SurfaceMaterial**. Using this boundary condition requires setting **FrequencyShift** in **EigenSolver** container close to the resonant mode frequency.
- **Absorbing**: Absorbing boundary condition (ABC) allows electromagnetic waves propagating at the speed of light to pass through the boundary without reflection. It is used for coaxial waveguides.
- **Waveguide**: Waveguide boundary condition to account for the dispersive propagation of electromagnetic wave in waveguides.
- **Periodic_M**: The master surface for periodic boundary condition; used together with **Periodic_S** and **Theta**.
- **Periodic_S**: The slave surface for periodic boundary condition; used together with **Periodic_M** and **Theta**. The surface meshes on the slave and master surfaces have to be exactly the same.
- **Theta**: The phase advance between the “Master” and “Slave” surfaces for periodic boundary condition; used together with **Periodic_M** and **Periodic_S**.

SurfaceMaterial specifies the external wall material properties, which are used to calculate the power loss on the surface.

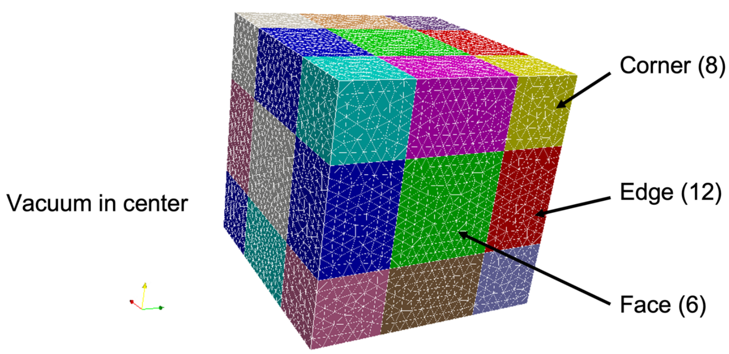
- **ReferenceNumber**: The reference number of the surface set in Cubit
- **Sigma**: Conductivity of the surface [unit in s/m]. If not specified, the exterior wall is copper with conductivity of 5.8e7 s/m.

Material specifies the material properties of a region in the model. If not specified, the region is set to vacuum. Every region of the mesh has a reference block number which is set in Cubit.

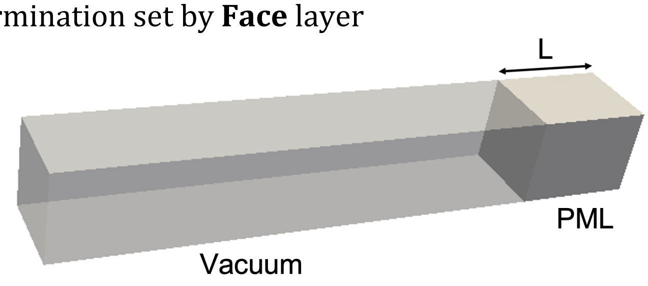
- **Attribute**: The ID of the region
- **Epsilon**: Relative permittivity of the region
- **Mu**: Relative permeability of the region
- **EpsilonImag**: Imaginary part of the relative permittivity of the region associated with material loss. The default is zero.
- **MuImag**: Imaginary part of the relative permeability of the region associated with material loss. The default is zero.
- **PML**: Container to specify perfectly matched layer (see setup below)

PML terminates wave propagation by absorption in artificially lossy layers.

- Free space termination set by **Face**, **Edge** and **Corner** layers



- Waveguide termination set by **Face** layer



- PML parameters specified by the following model

$$\sigma_w(l) = \sigma_{w,\max}\left(\frac{l}{d}\right)^m, \quad \sigma_{w,\max} = -\frac{(m+1)\ln R}{2\eta_0 d}$$

where s_w is the conductivity set by $s_{w,max}$, thickness *d*, order *m* and location in layer. *R* is the reflection coefficient from the PML.

PML container specifications

- **Type**: Type of PML layer

    - **Face**: 6 orientations specified by **Direction**
    - **Edge**: 12 orientations specified by **Direction**
    - **Corner**: 8 orientations specified by **Direction**
- **Start**: x, y, z
  Lower bound coordinates (x, y, z) of layer's bounding box [units in m]
- **End**: x, y, z
  Upper bound coordinates (x, y, z) of layer's bounding box [units in m]
- **Direction:** ix, iy, iz
  Orientation of layer:
    (+/- 1, 0, 0), (0, +/- 1, 0), (0, 0, +/- 1) for **Face** layer
    (+/- 1, +/- 1, 0), (0, +/- 1, +/- 1), (+/- 1, 0, +/- 1) for **Edge** layer
    (+/- 1, +/- 1, +/- 1) for **Corner** layer
- **Reflection Coefficient**: Reflected from PML; if specified, **Maximum Sigma** will be calculated.
- **Maximum Sigma**: Conductivity in x, y, z directions [units in S/m]

Spherical and cylindrical PML has slightly different specifications which requires the simulation model with a spherical or cylindrical layers. No bounding box and direction (**Start**, **End** and **Direction**) need to be specified. Additional parameters are

- **BoundaryID**: Reference number of PML outside surface
- **Thickness**: Spherical or cylindrical layer thickness [unit in m]
- **SubType**: Sphere for spherical PML; Cylinder for cylindrical PML; Note PML **Type** is **Face**.
- **Radius**: Radius of spherical or cylindrical PML [unit in m]

FiniteElement

```
FiniteElement:
{
  Order:          2
  CurvedSurfaces: on
}
```

FiniteElement specifies parameters for the finite element method.

- **Order**: The order of the finite elements used in the simulation. Order 1-6 have been implemented. The higher the order is, the more accurate the calculation is and the more computational resources are for a fixed mesh. For most applications, using 2nd order is good enough to obtain the solution accuracy.
- **Curved Surfaces**: **on** or **off**
  on: The surfaces of the finite elements on the model surface are represented by curved surfaces to better approximate the geometry. If no specified, curved surfaces are used.
  off: The surfaces of the finite elements on the model surface are represented by flat surfaces.

PRegion

```
FiniteElement:
{
 Order: 0
 CurvedSurfaces: on
}

PRegion:
{
 Type: Material
 Reference: 1
 Order: 2
}

PRegion:
{
 Type: Material
 Reference: 2
 Order: 3
}
```

PRegion enables to specify different orders of finite element basis functions (p) in different regions. Every region of the mesh has a reference block number which is set in Cubit. When using **PRegion**, the order used to be specified in the **FiniteElement** container is set to zero (as shown above) while that in every region of the computational domain has to be specified separately in **PRegion**.

- **Type**: The type of the region is **Material.**
- **Reference**: The reference number of the region
- **Order**: The finite element order

EigenSolver (Linear & Quadratic)

```
EigenSolver :
{
  NumEigenvalues: 1
  FrequencyShift:  1.0e9
  Tolerance: 1e-7
  MaxIterations: 300
}
```

EigenSolver specifies how many eigenmodes with frequency above a shifted frequency will be searched.

- **NumEignvalues**: The number of eigenmodes searched
- **FrequencyShift**: The frequency above which the eigenmodes are calculated [unit in Hz]
- **Tolerance:** Relative accuracy of the eigensolver
- **MaxIterations:** Maximum iteration, default is 200

The quadratic eigensolver solves both real and complex eigenvalues problems. For the complex eigenvalue problem, however, if the damping term is due to the waveguide loading, the port modes loaded at the ports need to be of the same cutoff. If the modes do not have the same cutoff frequencies, the eigenvalue problem becomes non-linear. Non-linear eigensolver (cork) needs to be used in order to solve such a problem.

Nonlinear EigenSolver

If the port modes loaded at a port do not have the same cutoff frequency, the eigenvalue problem becomes nonlinear. Nonlinear eigensolver (CORK) needs to be used in order to solve such a problem. There are additional parameters needed to be specified:

```
EigenSolver : {
 NumEigenvalues:  3
 Tolerance:      1e-10
 FrequencyShift:  1.0e9
 SolverMode:     CORK
 CORK: {
     TargetFrequencyMin: 1.0e9
     TargetFrequencyMax: 2.4e9
     TargetFrequencyImag: 5.0e8
     MaxIterations: 500
     ApproxTolerance: 1e-12
     ApproxMaxDegree: 100
 }
}
```

SolverMode: CORK

Invokes the nonlinear eigenvalue solver CORK.

- **TargetFrequencyMin**: The minimal frequency of the search region [unit in Hz]
- **TargetFrequencyMax**: The maximum frequency of the search region [unit in Hz]
- **TargetFrequencyImag**: The maximum of the imaginary component of the frequency [unit in Hz]
- **MaxIterations**: Maximum number of iterations for the nonlinear iteration
- **ApproxTolerance**: Tolerance of the nonlinear solver
- **ApproxMaxDegree**: Maximum degree of polynomial fit

Tips on using CORK:

```
     a) First run omega3p by loading enough modes at the ports.
     b) Check the 2D port modes in 2dports.log in job directory, e.g.,
omega3p_results.
     c) Compare the eigenmode frequencies and 2D port mode cutoff
frequencies and find out the number of propagating modes at each port for
the searched mode.
     c) Rerun omega3p by loading propagating modes only at each port to
eliminate numerical errors from the cutoff modes.
```

Port

```
Port :
{
 ReferenceNumber : 7
 NumberOfModes : 1
}
```

Port specifies the number of waveguide modes loaded on an open port. If not specified, only the lowest waveguide mode on the open port will be loaded.

- **ReferenceNumber**: The number should match the waveguide boundary ID.
- **NumberOfModes**: The number of modes loaded on the port

Absorbing Boundary at Port

If a port in the ModelInfo is specified as “absorbing”, this port will be treated with a simple 1D absorbing boundary condition (ABC). The cutoff frequency is zero (default). In this case the wave is assumed propagating with the speed of light, like plane wave in free space. One can, however, specify a cutoff frequency for the absorbing boundary condition to obtain a better approximation, as the follows:

```
Port : {
 ReferenceNumber : 7
 CutoffFrequency : 1e9
}
```

PostProcess

```
PostProcess :
{
 Toggle: on
 ModeFile: mode
 WriteLoss: on
}
```

Postprocess specifies postprocessing options.

- **Toggle**: **on** or **off**
  on: Mode files named *.mod will be created for the eigenmodes for visualization and analysis. This is the default option.
  off: No mode files named *.mod will be created for the eigenmodes.
- **ModeFile:** prefix of the mode file name. (default: omega3p)
- **WriteLoss**: **on** or **off**
  on: Mode files named *.loss will be created for power density distribution.
  off: No mode files named *.loss will be created for power density distribution [default].